**Claims**

We claim:

Sub A3

1. A method of creating a graphical program to perform an algorithm, the method comprising:

recording one or more functions in response to user input, wherein the one or more functions specify the algorithm; and

automatically generating the graphical program in response to the recorded one or more functions, wherein the graphical program implements the algorithm.

2. The method of claim 1, further comprising:

performing the one or more functions in response to user input;

wherein said recording the one or more functions is performed in response to said performing the one or more functions.

3. The method of claim 1,

wherein said recording the one or more functions comprises creating a prototype.

4. The method of claim 3,

wherein the prototype comprises a prototype in at least one of the disciplines from the group consisting of:

image processing, machine vision, image analysis, robotics, process control, industrial automation, test and measurement, simulation, workflow processes, and robotics.

5. The method of claim 1,

wherein said recording the one or more functions is performed in response to user input received via a graphical user interface (GUI).

6. The method of claim 5,

wherein the graphical user interface is associated with a prototyping environment application.

7. The method of claim 5,

wherein the user input comprises selecting the functions from one or more of a menu or palette.

8. The method of claim 1,

wherein said automatically generating the graphical program comprises programmatically generating the graphical program in response to the recorded one or more functions.

9. The method of claim 1,

wherein said automatically generating the graphical program comprises automatically generating graphical code in the graphical program without direct user input.

10. The method of claim 1, further comprising:

executing the graphical program to perform the algorithm.

11. The method of claim 1,

wherein the graphical program includes a block diagram portion and a user interface panel portion.

12. The method of claim 6,

wherein the graphical program is a graphical data flow program.

13. The method of claim 1,

wherein said automatically generating the graphical program comprises including one or more nodes corresponding to the one or more functions in the graphical program.

14. The method of claim 1, wherein the recorded one or more functions comprise a script, the method further comprising:

creating an association between the script and the graphical program;

modifying the script to create a new script in response to user input after said creating the association; and

modifying the graphical program according to the new script to create a new graphical program.

15. The method of claim 14,

wherein said modifying the graphical program according to the new script uses the association between the script and the graphical program;

wherein the association remains between the new script and the new graphical program.

16. The method of claim 14, further comprising:

receiving user input indicating a desire to change the graphical program;

displaying script information of the script;

modifying the script information in response to user input; and

modifying the graphical program after said modifying the script information.

17. The method of claim 1, further comprising:

creating an association between the script and the graphical program;

locking the association between the script and the graphical program, wherein said locking prevents user editing of the graphical program.

18. The method of claim 17, further comprising:

unlocking the association between the script and the graphical program in response to user input after said locking;

directly changing the graphical program in response to user input after said unlocking.

19. The method of claim 18,

wherein said unlocking removes the association between the script and the graphical program.

20. The method of claim 17, further comprising:

modifying the graphical program in response to user input after said generating the graphical program and after said creating the association between the script and the graphical program;

determining if an association exists between the script and the graphical program in response to said modifying the graphical program; and

removing the association between the script and the graphical program in response to said modifying.

21. The method of claim 1, further comprising:

receiving user input specifying code generation information;

wherein said automatically generating the graphical program utilizes the code generation information.

22. The method of claim 21,

wherein the code generation information specifies a type of graphical program to create in response to the recorded one or more functions;

wherein the graphical program is created in accordance with the specified graphical program type.

23. The method of claim 22,

wherein the graphical program type specifies a particular graphical programming environment;

wherein the graphical program is created in a file format that is usable by the particular graphical programming environment.

24. The method of claim 21,

wherein a plurality of parameters are associated with the one or more functions, wherein each parameter is an input parameter which provides input to a function or an output parameter which accepts output from a function;

wherein the code generation information specifies one or more of the input parameters which are desired to be interactively changeable or one or more of the output parameters which are desired to be interactively viewable;

wherein said automatically generating the graphical program comprises enabling the graphical program to receive user input during program operation, wherein the user input specifies values for the specified one or more input parameters;

wherein said automatically generating the graphical program comprises enabling the graphical program to display output during program operation, wherein the output indicates values for the specified one or more output parameters.

25. The method of claim 1,

wherein said automatically generating the graphical program comprises:

generating portions of graphical code, wherein each portion of graphical code implements one of the functions;

linking the portions of graphical code together.

26. The method of claim 25,

wherein each portion of graphical code includes one or more graphical program nodes, wherein each node has one or more inputs or outputs;

wherein generating each portion of graphical code comprises connecting the node inputs and outputs together in order to implement the function with which the portion of graphical code is associated.

27. The method of claim 26,

wherein linking a first portion of graphical code to a second portion of graphical code comprises connecting an output of a node in the first portion of graphical code to an input of a node in the second portion of graphical code.

28. The method of claim 26,

wherein at least one of the functions has an associated input parameter;

wherein each portion of code that implements a function that has an associated input parameter includes a node that has an input for receiving a value for the input parameter;

wherein each portion of code that implements a function that has an associated input parameter includes a leaf node that has an output for providing a value for the input parameter;

wherein the leaf node output for providing the parameter value is connected to the node input for receiving the parameter value.

29. The method of claim 26,

wherein at least one of the functions has an associated output parameter;

wherein each portion of code that implements a function that has an associated output parameter includes a node that has an output for providing a value for the output parameter;

wherein each portion of code that implements a function that has an associated output parameter includes a leaf node that has an input for receiving a value for the output parameter;

wherein the leaf node input for receiving the parameter value is connected to the node output for providing the parameter value.

30. The method of claim 25, further comprising:

for each function, retrieving information associated with the function from a database;

wherein generating the portion of graphical code that implements a particular function utilizes the database information retrieved for the particular function.

31. A system for creating a graphical program to perform an algorithm, the system comprising:

a processor;

a memory coupled to the processor which stores a prototyping environment application;

a user input device which receives user input;

wherein the prototyping environment application is executable in response to the user input to store one or more functions in the memory, wherein the one or more functions specify the algorithm;

wherein the prototyping environment application is executable to automatically generate a graphical program in response to the stored one or more functions, wherein the graphical program implements the algorithm specified by the one or more functions.

32. The system of claim 31,

wherein said storing the one or more functions in the memory comprises creating a prototype.

33. The system of claim 32,

wherein the prototyping environment application is a prototyping environment application in at least one of the disciplines from the group consisting of:

image processing, machine vision, image analysis, robotics, process control, industrial automation, test and measurement, simulation, workflow processes, and robotics.

34. The system of claim 31,

wherein the prototyping environment application includes a graphical user interface (GUI);

wherein said storing the one or more functions in the memory is performed in response to user input received via the graphical user interface.

35. The system of claim 34,

wherein the user input comprises selecting the functions from one or more of a menu or palette.

36. The system of claim 31, further comprising:

a graphical program creation program stored in the memory;

wherein the prototyping environment application is executable to call the graphical program creation program;

wherein the graphical program creation program is executable to automatically generate the graphical program in response to said prototyping environment application calling the graphical program creation program.

37. The system of claim 36,

wherein the graphical program creation program is a graphical programming development environment application.

38. The system of claim 31,

wherein said automatically generating the graphical program comprises programmatically generating the graphical program in response to the stored one or more functions.

39. The system of claim 31,

wherein said automatically generating the graphical program comprises automatically generating graphical code in the graphical program without direct user input.

40. The system of claim 31,

wherein the graphical program includes a block diagram portion and a user interface portion.

41. The system of claim 39,

wherein the graphical program is a graphical data flow program.

42. The system of claim 31,

wherein said automatically generating the graphical program comprises including one or more nodes corresponding to the one or more functions in the graphical program.

43. The system of claim 36,

wherein the stored one or more functions comprise a script;

wherein the memory stores an association between the script and the graphical program;

wherein the prototyping environment application is executable to utilize the association to modify the script to create a new script in response to user input;

wherein the graphical program creation program is executable to modify the graphical program according to the new script to create a new graphical program.

44. The system of claim 43,

wherein the memory stores an association between the script and the graphical program;

wherein the memory stores information specifying that the association between the script and the graphical program is locked, wherein said locking prevents user editing of the graphical program.

45. The system of claim 31,

wherein the prototyping environment application is executable to receive user input specifying code generation information, wherein the code generation specifies information to use in generating the graphical program.

46. The system of claim 45,

wherein the code generation information specifies a type of graphical program to create in response to the stored one or more functions;

wherein the graphical program is created in accordance with the specified graphical program type.

47. The method of claim 46,

wherein the graphical program type specifies a particular graphical programming environment;

wherein the graphical program is created in a file format that is usable by the particular graphical programming environment.

48. The system of claim 45,

wherein a plurality of parameters are associated with the functions, wherein each parameter is an input parameter which provides input to a function or an output parameter which accepts output from a function;

wherein the code generation information specifies one or more of the input parameters which are desired to be interactively changeable or one or more of the output parameters which are desired to be interactively viewable;

wherein said automatically generating the graphical program comprises enabling the graphical program to receive user input during program operation, wherein the user input specifies values for the specified one or more input parameters;

wherein said automatically generating the graphical program comprises enabling the graphical program to display output during program operation, wherein the output indicates values for the specified one or more output parameters.

49. The system of claim 31,

wherein said automatically generating the graphical program comprises:

generating portions of graphical code, wherein each portion of graphical code implements one of the functions;

linking the portions of graphical code together.

50. The system of claim 49,

wherein each portion of graphical code includes one or more graphical program nodes, wherein each node has one or more inputs or outputs;

wherein generating each portion of graphical code comprises connecting the node inputs and outputs together in order to implement the function with which the portion of graphical code is associated.

51. The system of claim 50,

wherein linking a first portion of graphical code to a second portion of graphical code comprises connecting an output of a node in the first portion of graphical code to an input of a node in the second portion of graphical code.

52. The system of claim 49,

wherein, for each function, information associated with the function is retrieved from a database;

wherein generating the portion of graphical code that implements a particular function utilizes the database information retrieved for the particular function.

53. A memory medium comprising program instructions executable to:

record one or more functions in response to user input, wherein the one or more functions specify an algorithm; and

automatically generate a graphical program in response to the recorded one or more functions, wherein the graphical program implements the algorithm.

54. The memory medium of claim 53, further comprising program instructions executable to:

perform the one or more functions in response to user input;

wherein said recording the one or more functions is performed in response to said performing the one or more functions.

55. The memory medium of claim 53,

wherein said recording the one or more functions comprises creating a prototype.

56. The memory medium of claim 55,

wherein the prototype comprises a prototype in at least one of the disciplines from the group consisting of:

image processing, machine vision, image analysis, robotics, process control, industrial automation, test and measurement, simulation, workflow processes, and robotics.

57. The memory medium of claim 53,

wherein said recording the one or more functions is performed in response to user input received via a graphical user interface (GUI).

58. The memory medium of claim 57,

wherein the graphical user interface is associated with a prototyping environment application.

59. The memory medium of claim 57,

wherein the user input comprises selecting the functions from one or more of a menu or palette.

60. The memory medium of claim 53,

wherein said automatically generating the graphical program comprises programmatically generating the graphical program in response to the recorded one or more functions.

61. The memory medium of claim 53,

wherein said automatically generating the graphical program comprises automatically generating graphical code in the graphical program without direct user input.

62. The memory medium of claim 53, further comprising program instructions executable to:

execute the graphical program to perform the algorithm.

63. The memory medium of claim 53,

wherein the graphical program includes a block diagram portion and a user interface panel portion.

64. The memory medium of claim 53,

wherein said automatically generating the graphical program comprises including one or more nodes corresponding to the one or more functions in the graphical program.

65. The memory medium of claim 53, wherein the recorded one or more functions comprise a script, the memory medium further comprising program instructions executable to:

create an association between the script and the graphical program;

modify the script to create a new script in response to user input after said creating the association; and

modify the graphical program according to the new script to create a new graphical program.

66. The memory medium of claim 65, further comprising program instructions executable to:

create an association between the script and the graphical program;

lock the association between the script and the graphical program, wherein said locking prevents user editing of the graphical program.

67. The memory medium of claim 53, further comprising program instructions executable to:

receive user input specifying code generation information;

wherein said automatically generating the graphical program utilizes the code generation information.

68. The memory medium of claim 67,

wherein the code generation information specifies a type of graphical program to create in response to the recorded one or more functions;

wherein the graphical program is created in accordance with the specified graphical program type.

69. The memory medium of claim 67,

wherein a plurality of parameters are associated with the one or more functions, wherein each parameter is an input parameter which provides input to a function or an output parameter which accepts output from a function;

wherein the code generation information specifies one or more of the input parameters which are desired to be interactively changeable or one or more of the output parameters which are desired to be interactively viewable;

wherein said automatically generating the graphical program comprises enabling the graphical program to receive user input during program operation, wherein the user input specifies values for the specified one or more input parameters;

wherein said automatically generating the graphical program comprises enabling the graphical program to display output during program operation, wherein the output indicates values for the specified one or more output parameters.

70. The memory medium of claim 53,

wherein said automatically generating the graphical program comprises:

generating portions of graphical code, wherein each portion of graphical code implements one of the functions;

linking the portions of graphical code together.

71. A method of creating a graphical program to perform an algorithm, the method comprising:

creating a prototype in response to user input, wherein the prototype specifies the algorithm; and

automatically generating the graphical program in response to the prototype, wherein the graphical program implements the algorithm.

72. The method of claim 71,

wherein the prototype comprises a prototype in at least one of the disciplines from the group consisting of:

image processing, machine vision, image analysis, robotics, process control, industrial automation, test and measurement, simulation, telecommunications, workflow processes, and robotics.

73. The method of claim 71,

wherein the user input is received via a graphical user interface (GUI) associated with a prototyping environment application.

74. The method of claim 73,

wherein the user input comprises selecting one or more functions from one or more of a menu or palette.

75. The method of claim 71,

wherein said automatically generating the graphical program comprises programmatically generating the graphical program in response to the created prototype.

76. The method of claim 71,

wherein said automatically generating the graphical program comprises programmatically generating graphical code in the graphical program without direct user input.

77. The method of claim 71,

wherein said automatically generating the graphical program comprises including one or more function nodes in the graphical program.

78. The method of claim 71,

wherein said creating the prototype in response to user input comprises creating a diagrammatic model of the algorithm.

79. The method of claim 71,

wherein said creating the prototype in response to user input comprises recording one or more functions in response to user input;

wherein the recorded one or more functions specify the algorithm.

80. The method of claim 79,

wherein said automatically generating the graphical program comprises:

generating portions of graphical code, wherein each portion of graphical code implements one of the functions;

linking the portions of graphical code together.

Add A87